Claims 1, 9, 16, 23, 33 and 41-44 are amended.

Claims 2, 12, 20, 27 and 35 are cancelled without prejudice.

Claims 1, 3-11, 13-19, 21-26, 28-34 and 36-44 remain in the Application as follows:

1. (Currently Amended) One or more tangible computer-readable media comprising a flash memory driver that is executable by a computer to interface between a file system and one or more flash memory media, the flash memory driver comprising:

flash abstraction logic that is invokable by the file system to manage flash memory operations without regard to the type of the one or more flash memory media; and

flash media logic configured to interact with different types of the flash memory media;

wherein the flash abstraction logic invokes the flash media logic to perform memory operations that are potentially performed in different ways depending on the type of the flash memory media, and further wherein the flash memory driver is flash memory medium agnostic, and wherein one of the flash memory operations includes performing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer.

2. (Cancelled).

3. **(Original)** The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes maintaining data integrity of the flash memory medium.

4. **(Original)** The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes handling recovery of data associated with the flash memory medium after a power-failure.

5. **(Original)** The flash memory driver as recited in Claim 1, wherein one of the flash memory operations includes mapping status information associated with physical sectors of the flash memory medium for use by the file system.

6. **(Previously Presented)** The flash memory driver as recited in Claim 1, wherein the flash medium logic is further configured to translate commands received from the file system to physical sector commands for issuance to the flash memory media.

7. **(Previously Presented)** The flash memory driver as recited in Claim 1, wherein the flash medium logic is user programmable to read, write and erase data to and from the flash memory media.

8. **(Previously Presented)** The flash memory driver as recited in Claim 1, wherein the flash media logic is configured to perform error code correction associated with the flash memory media.

9. (Currently Amended) A flash driver, comprising:

flash abstraction logic, interposed between a file system and a flash memory medium, configured to:

(a) map a logical sector status from the file system to a physical sector status of the flash memory medium; and

(b) maintain memory requirements associated with operating the flash memory medium;

wherein the flash driver is located remote from the flash memory medium, and wherein the memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer.

10. (Original) The flash driver as recited in Claim 9, further comprising a user programmable flash medium logic, configured to read, write and erase data to and from the flash memory medium.

11. (Original) The flash driver as recited in Claim 9, further comprising a user programmable flash medium logic configured to receive and translate specific operational commands from the file system associated with reading and writing data to the flash memory medium.

12. (Cancelled).

13. **(Original)** The flash driver as recited in Claim 9, wherein the memory requirements include maintaining data integrity of the flash memory medium.

14. **(Original)** The flash driver as recited in Claim 9, wherein the memory requirements include handling recovery of data associated with flash memory medium after a power-failure.

15. **(Original)** The flash driver as recited in Claim 9, further comprising a flash medium logic, programmably configurable by a user to perform error code correction associated with the flash memory medium.

16. **(Currently Amended)** A flash driver, comprising:

user programmable flash medium logic, configured to read, write and erase data to and from a flash memory medium; and

flash abstraction logic, interposed between a file system and flash memory medium to maintain universal requirements for the operation of the flash memory medium;

wherein the flash memory driver is flash memory medium agnostic, and wherein the universal requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer.

17. (Original) The flash driver as recited in Claim 16, wherein the flash abstraction logic passes specific commands associated with certain types of flash memory media directly to the flash medium logic for translation and execution.

18. (Original) The flash driver as recited in Claim 16, wherein the flash abstraction logic is an interface between the flash medium logic and the file system.

19. (Original) The flash driver as recited in Claim 16, wherein the universal requirements include maintaining data integrity of the flash memory medium.

20. (Cancelled).

21. (Original) The flash driver as recited in Claim 16, wherein the universal requirements include handling recovery after a power-failure.

22. (Original) The flash driver as recited in Claim 16, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with the type of flash memory medium selected.

23. (Currently Amended) A processing device that uses a flash memory medium for storage of data, comprising:

a file system, configured to control data storage for the processing device; flash media logic, configured to perform physical sector operations to a flash memory medium based on physical sector commands, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to interface with any type of flash memory medium selected; and

flash abstraction logic, configured to maintain flash memory requirements that are necessary to operate the flash memory medium, wherein the flash memory requirements include managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer.

24. (Original) The processing device as recited in Claim 23, wherein the flash abstraction logic passes physical logic commands associated with certain types of flash memory medium directly to the flash memory medium logic for translation and execution.

25. (Original) The processing device as recited in Claim 23, wherein the flash abstraction logic is an interface between the flash medium logic and the file system.

26. (Original) The processing device as recited in Claim 23, wherein the flash memory requirements include maintaining data integrity of the flash memory medium.

27. (Cancelled).

28. (Original) The processing device as recited in Claim 23, wherein the flash memory requirements include handling recovery after a power-failure.

29. (Original) The processing device as recited in Claim 23, wherein the requirements are common to a plurality of different flash memory media.

30. (Original) The processing device as recited in Claim 23, wherein the flash medium logic comprises a set of programmable entry points that can be implemented by a user to perform error code correction with the type of flash memory medium used in the processing device.

31. (Original) The processing device as recited in Claim 23, whereby the flash medium logic relieves the flash abstraction logic from performing translation of the physical sector commands received from the file system.

32. (Original) The processing device as recited in Claim 23, wherein the physical sector operations include read, write and error code correction commands associated with the flash memory medium.

33. (Currently Amended) In a processing device that uses a flash memory medium for storage of data, a method for driving the flash memory medium, comprising:

managing rules associated with operating the flash memory medium in a flash abstraction logic; and

issuing physical sector commands directly to the flash memory medium from a flash medium logic;

wherein the method is flash memory medium agnostic, and wherein one of the rules includes managing wear-leveling operations associated with the flash memory medium by way of circular and continuous advancement of a write pointer.

34. (Original) The method as recited in Claim 33, wherein one of the rules includes maintaining data integrity of the flash memory medium.

35. (Cancelled).

36. (Original) The method as recited in Claim 33, wherein one of the rules includes handling recovery of the media after a power-failure.

37. (Original) The method as recited in Claim 33, wherein issuing physical sector commands directly to the flash memory medium comprises receiving read and write commands from a file system and translating them into the physical sector commands.

38. (Original) The method as recited in Claim 33, further comprising issuing a set of programmable entry points that can be implemented by a user to perform error code correction with the type of flash memory medium used in the processing device.

39. (Original) The method as recited in Claim 33, further comprising issuing a set of programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium used in the processing device.

40. (Original) The method as recited in Claim 33, further comprising receiving read and write commands from a file system.

41. (Currently Amended) One or more <u>tangible</u> computer-readable media comprising computer-executable instructions that, when executed, perform the method as recited in claim 33.

42. (Currently Amended) A <u>tangible</u> computer-readable medium for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to provide an interface between a file system, selected from one of a plurality of different file systems, and a flash memory medium, selected from one of a plurality of different flash memory media, wherein the flash driver is located remote from the flash memory medium<u>, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer</u>.

43. (Currently Amended) A tangible computer-readable medium for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media; and

manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic;

wherein the flash driver is flash memory medium agnostic, and wherein wear-leveling of the flash memory medium is performed by way of circular and continuous advancement of a write pointer.

44. (Currently Amended) A tangible computer-readable medium for a flash driver, comprising computer-executable instructions that, when executed, direct the flash driver to:

provide an interface between a file system, selected from one of a plurality of different files systems, and a flash memory medium, selected from one of a plurality of different flash memory media;

manage a set of characteristics that are common to the plurality of different flash memory media at a flash abstraction logic; and

provide programmable entry points that can be optionally selected by a user to interface with the type of flash memory medium selected;

wherein the flash driver is located remote from the flash memory medium and the flash driver is flash memory medium agnostic, and wherein wear-leveling of a flash memory medium is performed by way of circular and continuous advancement of a write pointer.